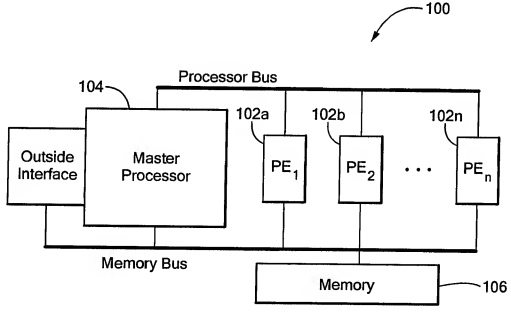
100
Processor Bus
104
102a
102b
102n
Outside Interface
Master Processor
PE1
PE2
PEn
Memory Bus
Memory
106

FIG. 1

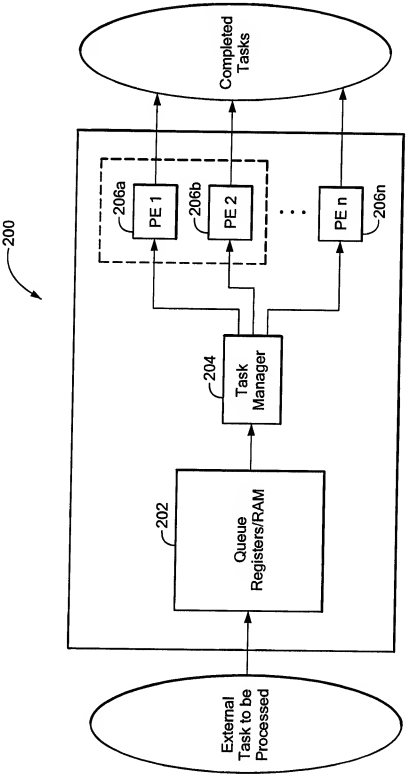
200
External Task to be Processed
202
Queue Registers/RAM
204
Task Manager
206a
PE 1
206b
PE 2
PE n
206n
Completed Tasks

FIG. 2A

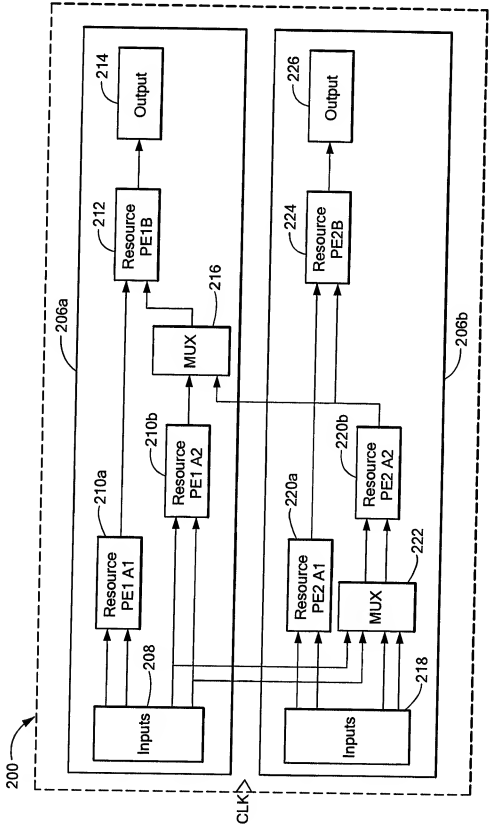
200
206a
206b
214
Output
212
Resource PE1B
216
MUX
210a
Resource PE1 A1
210b
Resource PE1 A2
208
Inputs
CLK
226
Output
224
Resource PE2B
220a
Resource PE2 A1
220b
Resource PE2 A2
222
MUX
218
Inputs

FIG. 2B

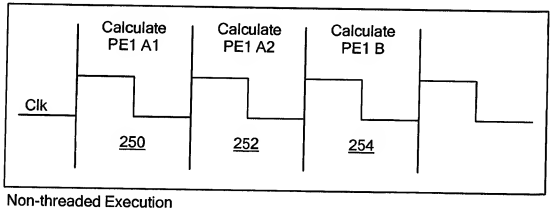
Calculate
PE1 A1
Calculate
PE1 A2
Calculate
PE1 B
Clk
250
252
254
Non-threaded Execution

FIG. 2C

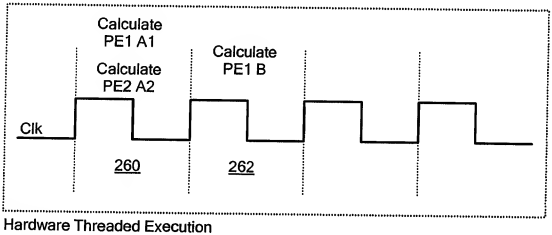
Calculate
PE1 A1
Calculate
PE2 A2
Calculate
PE1 B
Clk
260
262
Hardware Threaded Execution

FIG. 2D

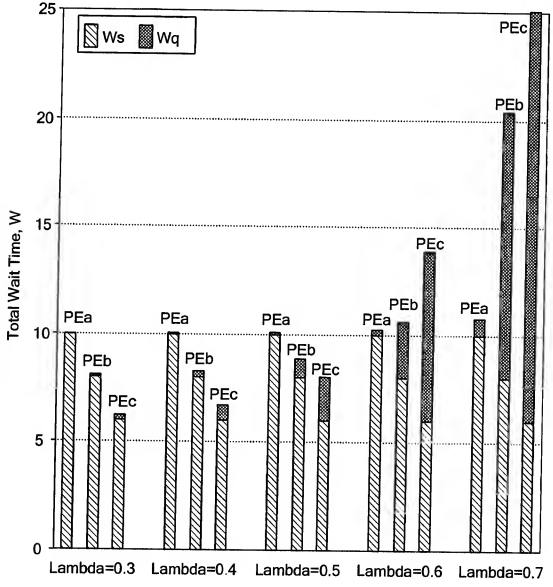
Ws
Wq
Total Wait Time, W
25
20
15
10
5
0
PEa
PEb
PEc
Lambda=0.3
Lambda=0.4
Lambda=0.5
Lambda=0.6
Lambda=0.7

FIG. 3

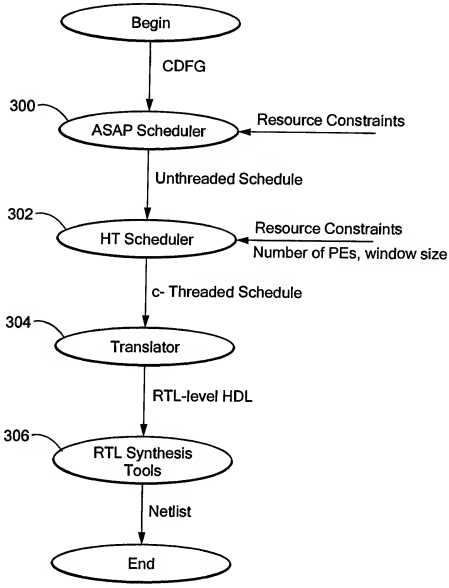
Begin
CDFG
300
ASAP Scheduler
Resource Constraints
Unthreaded Schedule
302
HT Scheduler
Resource Constraints
Number of PEs, window size
c- Threaded Schedule
304
Translator
RTL-level HDL
306
RTL Synthesis Tools
Netlist
End

FIG. 4

FIG. 5A

FIG. 5B

FIG. 5C

FIG. 5D

HTscheduler *(S,R,n,w)*

Input: $S \equiv$ Unthreaded Schedule
$R \equiv$ Resource Constraints Per Partition
$n \equiv$ Number of Partitions
$w \equiv$ Window Size

Vars: *CurrentState* $\equiv$ Current State to be Scheduled
NextState $\equiv$ Next State to be Scheduled
ByPassedEdges $\equiv$ Queue of Edges to Bypass
Borrowed $\equiv$ Queue of Borrowed States
StatesAlreadyScheduled $\equiv$ Array of states already scheduled

Output: *HTS* $\equiv$ Hardware-threaded Schedule

```
1.  CurrentState = initialState(S);
2.  while Current State is not null do
3.    if IsConditional(CurrentState) then
4.     for each child of CurrentState
5.       HTscheduler(Schedule(child),R,n,w)
6.    else if StatesAlreadyScheduled(CurrentState) =0 then
7.       StatesAlreadyScheduled(CurrentState) =1
8.       NextState=CurrentState
9.       for ( i=0;≤w;i++) do
10.         NextState=successor(NextState)
11.         if IsConditional(NextState) v IsJoining(NextState) then
12.           i=w
13.         else
14.           if IsSchedulable(CurrentState,NextState,R,n) then
15.             ByPassedEdges.append(outedge(NextState))
16.             Borrowed.append(NextState)
17.             StateAlreadyScheduled(NextState) =1
18.             CombineStates(CurrentState,NextState)
19.   if successor(CurrentState)=Borrowed.top then
20.     Borrowed.dequeue.top
21.     while ! IsEmpty(Borrowed) ∧
                Borrowed.top=(target(ByPassedEdges.top)) do
22.       Borrowed.dequeuetop
23.       ByPassedEdges.dequeuetop
24.     CreateEdge(CurrentState,target(ByPassedEdges.top),dashed)
25.     ByPassedEdges.dequeuetop
26.   else
27.     CreateEdge(CurrentState,successor(CurrentState),dashed)
28.   CurrentState=successor(CurrentState)
```

FIG. 6

FIG. 7A

FIG. 7B

FIG. 7C

FIG. 7D

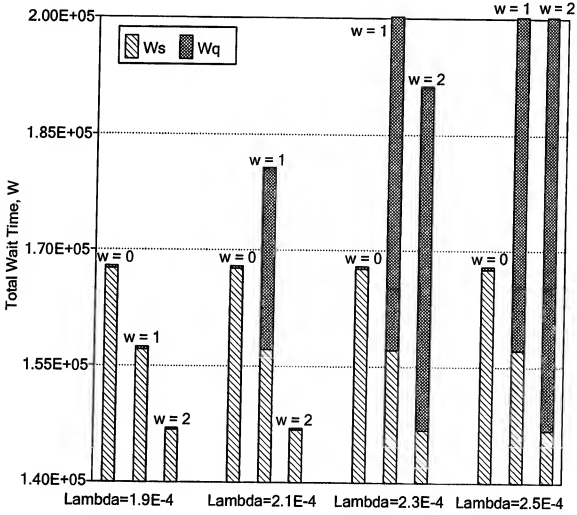
2.00E+05
1.85E+05
1.70E+05
1.55E+05
1.40E+05
Total Wait Time, W
Ws
Wq
w = 0
w = 1
w = 2
Lambda=1.9E-4
Lambda=2.1E-4
Lambda=2.3E-4
Lambda=2.5E-4

FIG. 8

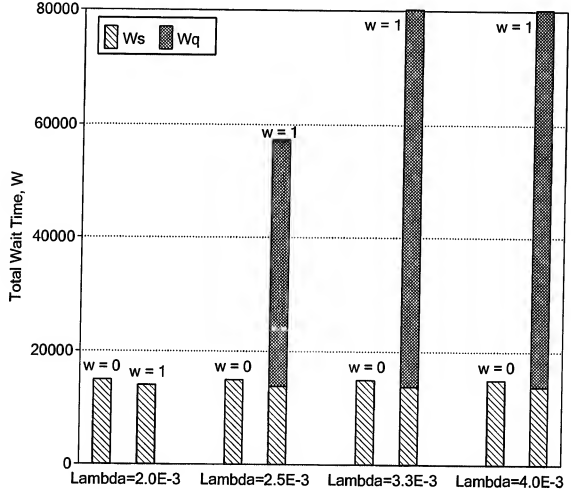
80000
60000
40000
20000
0
Total Wait Time, W
Ws
Wq
w = 0
w = 1
w = 0
w = 1
w = 0
w = 1
w = 0
w = 1
Lambda=2.0E-3
Lambda=2.5E-3
Lambda=3.3E-3
Lambda=4.0E-3

FIG. 9

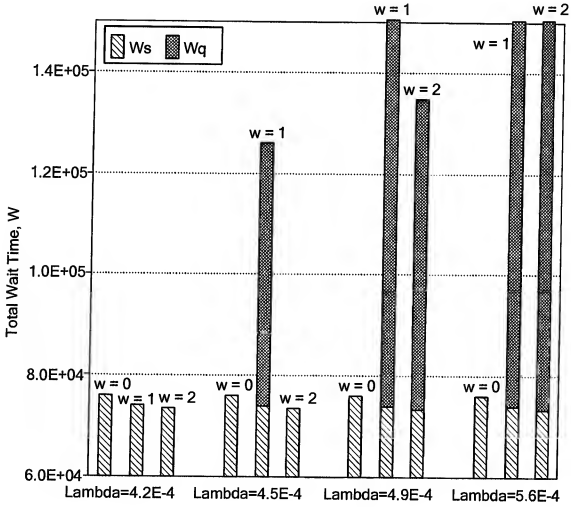
Ws
Wq
Total Wait Time, W
1.4E+05
1.2E+05
1.0E+05
8.0E+04
6.0E+04
w = 0
w = 1
w = 2
Lambda=4.2E-4
Lambda=4.5E-4
Lambda=4.9E-4
Lambda=5.6E-4

FIG. 10

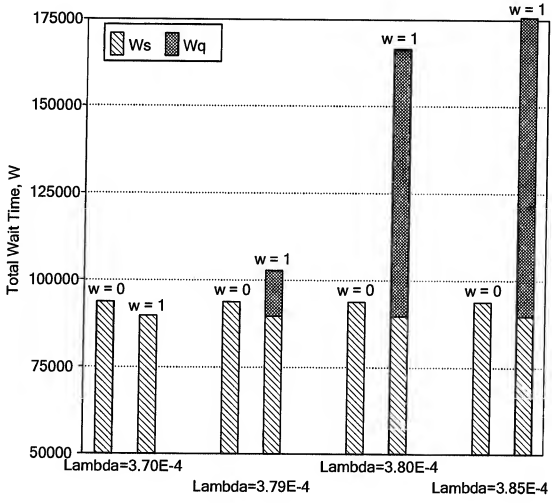
Total Wait Time, W
175000
150000
125000
100000
75000
50000
Ws
Wq
w = 0
w = 1
w = 0
w = 1
w = 0
w = 1
w = 0
w = 1
Lambda=3.70E-4
Lambda=3.79E-4
Lambda=3.80E-4
Lambda=3.85E-4

FIG. 11